

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

Bakalářská práce

2012

Gabriel Bischof

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Online hra pro výuku databázových
systémů
Online Game to Support Database Subjects

2012

Gabriel Bischof

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student:

Gabriel Bischof

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Online hra pro výuku databázových systémů
Online Game to Support Database Subjects

Zásady pro vypracování:

Náplní práce je vytvořit on-line hru dle zadání specifikovaném v [1]. Tato hra bude sloužit k procvičování základních pojmů z oblasti databází. Hrací princip je postaven na pohybu hráče bludištěm, kde musí odpovídat na kvízové otázky spojené s databázemi.

Hlavními body, jež musí autor splnit, jsou:

1. Vytvoření on-line hry dle specifikace s využitím vybraných technologií.
2. Vytvoření administrativního modulu, jež bude možné použít pro definování nových kol a otázek.

Při řešení bude student postupovat takto:

1. Výběr vhodné technologie pro danou aplikaci.
2. Analýza a návrh aplikace.
3. Implementace a podrobné otestování.

Seznam doporučené odborné literatury:

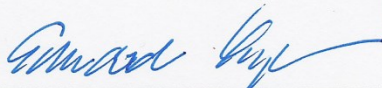
[1] Radim Bača. Podrobný popis herního systému pro výuku databází.
<http://homel.vsb.cz/~bac027/zadani/databaze.pdf>

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí bakalářské práce: **Ing. Radim Bača, Ph.D.**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne: 17.8.2012 .v. Osobně.....

Podpis 

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce, Ing. Radimu Bačovi, Ph.D., za odbornou pomoc a konzultaci při vytváření této práce.

Abstrakt

Cílem této práce je vytvoření webové hry. Tato hra bude sloužit k výuce a k procvičování základních pojmů z oblasti databází. Hrací princip je postaven na pohybu hráče bludištěm, v kterém musí odpovídat na kvízové otázky spojené s databázemi. Pro vytvoření této práce bylo použito technologie Microsoft Silverlight, ASP.NET, WCF a jazyka C#. Databáze MS SQL byla zvolena pro uložení definičních a herních údajů. Pomocí technologie Silverlight a jazyka C# v pozadí byla vytvořena GUI a logika hry. Komunikaci s databází zajišťuje služba WCF. Pro administraci dat uložených v databázi slouží samostatná online aplikace vytvořená pomocí technologie ASP.NET.

Klíčová slova

bakalářská práce, Silverlight, C#, ASP.NET, MS SQL, WCF, .NET

Abstract

The aim of the bachelor work is creation of web-based game. This game will be used for teaching and practice of the basic concepts of databases. Playing principle is based on player's movement through the maze, where he has to answer the quiz questions related to databases. This work was created using technologies Microsoft Silverlight, ASP.NET, WCF and C#. Gaming data and definitions are saved in MS SQL database. GUI and game logic was created in Silverlight with C# in codebehind. Communication with database is handled by WCF service. Data in database are managed in standalone online application created in ASP.NET.

Key words

bachelor's thesis, Silverlight, C#, ASP.NET, MS SQL, WCF, .NET

Seznam použitých symbolů a zkratek

ActionScript	objektově orientovaný programovací jazyk pro aplikace vyvíjené pomocí Flashe
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
ASP.NET	součást .NET Frameworku pro tvorbu webových aplikací a služeb
Canvas	element jazyka XAML, umožňující dynamické vykreslování komponent
connectionString	řetězec obsahující parametry nezbytné pro připojení aplikace k databázi
Dapper	micro-ORM pro technologii .NET
DB	Database
Entity Framework	nový typ datasource v asp.net aplikacích které jsou postaveny nad asp.net 4
Flash	grafický vektorový program, používá se především pro tvorbu (převážně internetových) interaktivních animací, prezentací a her
Framework	Softwarová struktura, která slouží jako podpora při programování, vývoji a organizaci jiných softwarových projektů. Může obsahovat podpůrné programy, knihovnu API, návrhové vzory nebo doporučené postupy při vývoji.
Game loop	herní smyčka
Gridview	Obecně definovaná komponenta, která zobrazuje data v tabulkovém formátu, a u každého záznamu mohou být přidána funkční tlačítka pro manipulaci se záznamem. Data mohou být pro větší přehlednost rozložena do stránek.
GUI	Graphic User Interface
HTML	Hyper Text Markup Language
HTML5	rozšiřující specifikace jazyka HTML
HTTP	Hypertext Transfer Protocol
IDE	Integrated Developement Environment
JavaScript	Dynamický multiplatformní skriptovací jazyk.
JIT	Just In Time

LINQ	Language Integrated Query; integrovaný jazyk pro dotazování v technologii .Net 3.5 a vyšší
MS SQL	relační databázový systém firmy Microsoft, založený na SQL
ORM	Objektově relační mapování
RIA	webová aplikace, která má některé vlastnosti grafické desktopové aplikace
SDK	Software Development Kit
Silverlight	Rich Internet Application; multiplatformní aplikační platforma vytvořená společností Microsoft, která je určena pro vývoj business a multimediálních aplikací (RIA)
SQL	Structured Query Language (strukturovaný dotazovací jazyk)
SQL Injection	technika napadení databázové vrstvy programu vsunutím (odtud "injection") kódu přes neošetřený vstup a vykonání vlastního, samozřejmě pozměněného, SQL dotazu
SWF	přípona souborů které jsou interpretovány pomocí aplikace Adobe Flash Player
TCP/IP	Transmission Control Protocol/Internet Protocol; Rodina protokolů, obsahuje sadu protokolů pro komunikaci v počítačové síti a je hlavním protokolem celosvětové sítě Internet
UI	User Interface
XAML	Extensible Application Markup Language
XAP	přípona souborů Silverlight aplikací
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language
WCF	framework pro webové služby

Seznam tabulek

Tabulka 1: USER - tabulka s uživateli	- 4 -
Tabulka 2: SCORE - tabulka s výsledky	- 4 -
Tabulka 3: ROUND - tabulka s atributy kola	- 4 -
Tabulka 4: FIELD - tabulka s jednotlivými políčky kola	- 4 -
Tabulka 5: QUESTION - tabulka znění otázek	- 5 -
Tabulka 6: ANSWER - tabulka možných odpovědí na otázky	- 5 -

Seznam obrázků

Obrázek 1: Analýza koncepce herního systému.....	10
Obrázek 2: DFD diagram 0. Úrovně	10
Obrázek 3: Zjednodušený ER diagram.....	10
Obrázek 4: DFD diagram 1. Úrovně zobrazující akce přístupné uživateli s administrátorskými právy	11
Obrázek 5: DFD diagram 2. Úrovně pro administraci kol (editace a mazání)	12
Obrázek 6: DFD diagram 2. Úrovně pro administraci kol (vlození).....	12
Obrázek 7: DFD diagram 2. Úrovně pro administraci výsledků	13
Obrázek 8: Use-case diagram administrace	13
Obrázek 9: Use-case diagram hry.....	14
Obrázek 10: Stavový diagram hry	15
Obrázek 11: Stavový diagram otázky.....	16
Obrázek 12: Náhled na celé řešení aplikace.....	17
Obrázek 13: Use-case diagram aplikace.....	18
Obrázek 14: ER diagram	19
Obrázek 15: Znázornění herní smyčky.....	20
Obrázek 16: Příklad kontroly viditelnosti	21
Obrázek 17: Náhled na rozhraní administračního modulu – návrh kola.....	26

Obsah

1	Úvod	3
2	Porovnání technologií	4
2.1	Flash	4
2.1.1	ActionScript	4
2.2	Silverlight	5
2.2.1	XAML	5
2.3	HTML5	6
2.4	Porovnání Silverlight a Flash	6
2.5	Proč jsem se rozhodl využít technologii Silverlight	7
3	Princip hry	8
4	Analýza	10
4.1	Administrace	11
4.2	Hra	13
5	Návrh	17
5.1	Náhled na celé řešení	17
5.2	Uložení dat	18
5.3	Netriviální algoritmy	20
5.3.1	Herní smyčka	20
5.3.2	Viditelnost mezi strážcem a hráčem	21
5.3.3	Hledání nejkratší možné trasy v bludišti.	22
5.4	Silverlight aplikace	23
5.5	WCF služba	23
5.6	Administrace	24
5.6.1	Administrace uživatelů	24
5.6.2	Administrace výsledků	25
5.6.3	Administrace otázek	25
5.6.4	Administrace kol	25
6	Řešení	27
6.1	Popis aplikace	27
6.2	Nasazení aplikace	27

7	Závěr.....	29
	Příloha č. 1: Datový slovník	- 4 -

1 Úvod

V dnešní době nás počítače a jejich využití provází na každém kroku, a proto je kladen důraz na to, aby s nimi uměla pracovat široká společnost. Základy počítačové gramotnosti ovládají již studenti základních škol. Existují předměty zaměřené nejen na výuku práce s počítačem, ale i mnohé předměty, k jejichž výuce jsou počítače se specializovaným softwarem využívány. Díky tomu výuka probíhá poutavou a hravou formou.

Tento typ výuky je stále více využíván, a proto jsem se rozhodl zaměřit právě na tuto tematiku. Cílem mé práce je tedy vytvořit výukovou pomůcku pro ověřování znalostí nabytých v databázových předmětech na vysoké škole.

Pro odlehčení tematiky byla pro výukovou aplikaci zvolena forma jednoduché hry. V této práci budu popisovat celý průběh vývoje této aplikace. Tento proces začíná výběrem vhodných technologií a analýzou požadavků na aplikaci. Následně pokračuje návrhem řešení a končí implementací samotné aplikace.

2 Porovnání technologií

Webová hra je online počítačová hra, kterou hráči ovládají pouze prostřednictvím standartního webového prohlížeče. Výhodou takových her je jednoduchá dosažitelnost (Lze hrát prakticky na jakémkoli počítači s připojením k internetu).

Nejjednodušší hry fungují na čistém (X)HTML s grafickou stránkou doplněnou obrázky a kaskádovými styly. Jelikož je HTML statické, je tento způsob realizace vhodný spíše pro tahové hry, ve kterých neprobíhá akce v reálném čase. Rozšíření funkcionality lze snadno dosáhnout s využitím klientského Javascriptu, který umožňuje pohodlnější ovládání a lepší aktualizaci některých informací, zvláště při využití technologie AJAX. Všechny doposud uvedené technologie jsou podporovány již v základní instalaci všech moderních běžně používaných prohlížečů, takže hry lze hrát opravdu téměř na každém počítači.

Jako webové hry však lze označit také o něco pokročilejší hry. Ty se sice ovládají prostřednictvím prohlížeče, ale vyžadují instalaci nějakého rozšiřujícího programu. K těmto rozšiřujícím technologiím patří zejména Java, Flash a Silverlight. V poslední době se však stále více mluví o HTML5, jež je rozšiřující specifikací HTML. Cílem HTML5 je umožnit vývoj pokročilejších webových aplikací bez nutnosti použití zmíněných rozšiřujících technologií.

2.1 Flash

Flash je grafický vektorový program. Byl vyvinut firmou Macromedia, avšak tuto firmu a všechny její produkty nyní vlastní společnost Adobe. Pomocí této technologie lze vytvářet interaktivní animace, prezentace, reklamy a webové hry. Jelikož jsou informace ukládány ve vektorovém formátu, dosahují výsledné soubory poměrně malých velikostí. Díky tomu se Flash stal velice oblíbeným webovým prvkem. Protože do uvedení technologie Silverlight neexistovala jiná, Flashi konkurenceschopná technologie, získal si Flash pozici nepoužívanější technologie pro interaktivní prvky.

Flash umožňuje export do dvou základních formátů:

- swf - v tomto formátu má soubor malou velikost, může být přehráván ve webovém prohlížeči, ale k jeho běhu je nutný přehrávač Adobe Flash Player, který je volně ke stažení na stránkách společnosti Adobe.
- exe - formát určený pro spuštění ve Windows bez nutnosti použití dalšího přehrávače. Soubor má však větší velikost, jelikož má v sobě již implementovaný Flash Player.

2.1.1 ActionScript

Flash má také vlastní implementovaný programovací jazyk ActionScript. V současné verzi 3.0 je ActionScript poměrně vyspělý objektově orientovaný jazyk, obsahující jak základní datové typy, tak komplexní datové typy pro práci s grafikou, zvukem, externími soubory (např. XML) nebo síťovou komunikaci. ActionScript využívá JIT kompilaci, jež přináší výhody například při využití tabletů a chytrých telefonů, které mají omezenou paměť.

2.2 Silverlight

Microsoft Silverlight je multiplatformní implementace .NET Frameworku, určená pro tvorbu aplikací RIA a mediálních prezentací na webu. Jedním z důvodů vzniku technologie Silverlight byla přímá konkurence technologii Adobe Flash ze strany společnosti Microsoft.

K nejsilnějším stránkám Silverlightu je přehrávání videa a velké množství funkcí, které je možné využít při tvorbě business aplikací.

První verze Silverlightu byla představena pod názvem Windows Presentation Foundation/Everywhere v roce 2007. Během jednoho roku proběhl prudký vývoj této technologie až na verzi 3.0, která již umožňovala použití programovacích jazyků C#, VB.NET, JavaScript a IronPython. Jako zlomová se jeví verze 4.0, jež nabídla například plnou podporu tisku, funkce drag'n'drop, použití všech tlačítek myši a zdokonalila funkci Out-of-Browser, jež umožňuje spouštět aplikace bez použití prohlížeče a dokonce i v režimu celé obrazovky.

Současná verze 5 byla vydána v prosinci 2011. Tato verze přinesla podporu 64-bitových systémů a hardwarovou akceleraci přehrávání videa ve vysokém rozlišení. Tato verze je zřejmě poslední vydanou hlavní verzí technologie, protože společnost Microsoft vidí budoucnost multiplatformních aplikací v HTML5. To však neznamená, že by byl Silverlight mrtvý. Verze 5 bude podporována až do roku 2021, jedná se tedy o jakousi přechodovou technologii, do té doby než bude HTML5 dostatečně vyspělé.

2.2.1 XAML

Extensible Application Markup Language (XAML) byl představen současně s technologií WPF, na které je založen systém Windows Vista a je na ní stavěn i operační systém Windows Seven.

XAML je deklarativní programovací jazyk, syntakticky vycházející z jazyka XML. Skládá se tedy s jednotlivých elementů, jejichž názvy (tagy) se zapisují v ostrých závorkách. Tagy se vyskytují buď samostatně - tzv. selfclosing, tedy samouzavírací nebo ve dvojicích, kde jeden je počáteční a druhý je uzavírací. Pokud se nějaké elementy vyskytují mezi počátečním a uzavíracím tagem jiného elementu, pak tvrdíme, že tyto elementy jsou potomci či subelementy onoho nadřazeného elementu, jenž je jejich rodičem. Velmi užitečnou vlastností jazyka XAML je fakt, že naprostou většinu parametrů jednotlivých elementů lze použít jako jejich subelementy. Např. chceme-li nastavit barvu pozadí obdélníku, můžeme to udělat takto:

```
<Rectangle Fill="Blue"/>
```

Nebo takto:

```
<Rectangle>
  <Rectangle.Fill>
    <SolidColorBrush Color="Blue"/>
  </Rectangle.Fill>
</Rectangle>
```


Díky tomuto způsobu zadávání parametrů je možné vytvořit mnohem komplexnější styl pozadí (lineární interpolace, obrázek, videoklip, atd.) velmi jednoduše a přehledně. Stejným způsobem lze např. na tlačítko umístit místo základního popisku obrázek nebo jakýkoli jiný objekt, popř. i více objektů najednou.

Jazyk XAML je nedílnou součástí technologie Silverlight, jelikož s jeho pomocí se vytváří grafický design Silverlight aplikací (stejně jako v případě WPF).

2.3 HTML5

HTML5 je nová multiplatformní technologie pro vývoj webových prvků včetně RIA aplikací. HTML5 vzniklo jako reakce na neúspěch XHTML a na to, že dnešní HTML4 potřebuje na složitější pohyblivé prvky a video zásuvné moduly - zejména právě Flash a Silverlight.

Bohužel je HTML5 stále ve fázi vývoje, dokonce ještě není dokončena ani jeho specifikace [5]. Další problém je v současných verzích prohlížečů, jež HTML5 zatím plně nepodporují.

2.4 Porovnání Silverlight a Flash

Největší rozdíl mezi Silverlightem a Flashem nalezneme nejspíše v systému animace. Flash poskytuje tzv. frame-by-frame animaci. To znamená, že pro každou změnu stavu určitého objektu na scéně za jednotku času je vypočítáno, jak přesně bude daný objekt vypadat na všech snímcích, které tuto změnu zobrazují. Bohužel to znamená, že má-li uživatel pomalejší počítač, pak může zobrazení každého jednotlivého snímku trvat delší dobu než by mělo. Výsledkem je prodloužení času, který je na animaci vyhrazen. Oproti tomu Silverlight pracuje na principu stavové animace. Tento princip vyžaduje pouze určení počátečního a finálního stavu objektu a času, během kterého má být změna provedena. Silverlight sám dokáže zjistit, zda bude výkon počítače na zobrazení stačit, a přizpůsobit příslušné mezistavy objektu tomu, aby změna opravdu proběhla v zadaném čase.

Obecně by se dalo říci, že technologie Silverlight je v mnohém jednodušší a pro vývojáře příjemnější než Flash, jelikož hodně práce udělá za něj. Oproti tomu technologie Flash svou volností (většinu práce musí obstarat sám vývojář) umožňuje vytvářet graficky i funkčně unikátní aplikace, zatímco Silverlight je odkázán na grafické možnosti jazyka XAML.

Stěžejním předmětem porovnání je také podpora video formátů. Technologie Silverlight totiž nabídla už v první verzi možnost přehrávání videa v HD kvalitě, zatímco Flash tuto vlastnost neměl. Konkrétním kodekovým standardem, který umožňuje technologii Silverlight přehrávat HD video je VC-1. Dnes již společnost Adobe odpověděla na tuto novinku tím, že do Flash Playeru začlenila podporu standardu H.264, také umožňujícího přehrávání videa v HD rozlišení.

Nespornou výhodou technologie Silverlight je spojení s platformou .NET. Díky tomu lze totiž využít hned několika programovacích jazyků, je k dispozici velmi široká nabídka knihoven a komponent a to vše se používá velmi podobně, jako kdybychom vytvářeli Windows aplikace. K Silverlight existuje celá řada volně stažitelných komponent, hlavní výhodou však není to, že si lze spoustu komponent

stáhnout, ale samotný fakt, že je velmi jednoduché si vlastní komponenty vytvářet pomocí jazyka XAML.

2.5 Proč jsem se rozhodl využít technologii Silverlight

Rozhodl jsem se využít technologie Microsoft Silverlight, protože jsem již měl základní znalosti jazyka C# a knihoven .NET. Jelikož mě tyto technologie velice zaujaly, viděl jsem v jejich použití příležitost se nimi blíže seznámit. Další výhodou Silverlightu je existence velice komfortních vývojových prostředí – Microsoft Visual Studio pro kód v pozadí a Microsoft Expression Blend pro GUI.

3 Princip hry

Pro zvýšení herní atraktivity a silnější motivaci hráče, není cílem hry pouze správné zodpovězení položených otázek a úspěšné dokončení veškerých kol, ale je zde aplikováno zpestření ve formě strážců, bonusů a bonusových bodů za úspěšné dokončení kola v co nejkratším čase. Existuje-li jiná cesta k východu z bludiště, není zapotřebí odpovédět na veškeré otázky správně, neboť to není pro úspěšné dokončení kola podmínkou. Ovšem při špatném zodpovězení položené otázky je hráč vrácen na startovní pozici v bludišti a automaticky také ztrácí jeden ze svých životů. Hráče však k odpovídání na otázky motivuje možnost získání bodů a bonusů. Tyto bonusy mu pak usnadňují další postup ve hře.

Strategická místa bludiště budou hlídána strážci a hráč bude muset využít získané bonusy ve formě zbraní k odstranění těchto strážců.

Strážců se ve hře vyskytuje několik typů –

- Typ 1 - začne pronásledovat jakmile je hráč ve střežené oblasti, která je určena tvůrcem kola a strážce má přímou viditelnost na hráče. Tento strážce je pomalejší než hráč.
- Typ 2 - začne pronásledovat pokud je hráč ve střežené oblasti i bez přímé viditelnosti na hráče. Je stejně rychlý jako hráč.
- Typ 3 - stejné jako typ 2, avšak hodnocen více body (Bohužel se nepodařilo splnit zadání, podle kterého se měl strážce vyhýbat střelám)
- Typ 4 - stejné jako typ 3, avšak hodnocen více body (Bohužel se nepodařilo splnit zadání, podle kterého se měl strážce vyhýbat střelám a mít více životů)

Při zabití strážcem platí podmínky uvedené již výše, tedy hráč je opětovně vrácen na startovní pozici v bludišti a automaticky také ztrácí jeden ze svých životů. Každé herní kolo má přesně definováno čas, za který musí být dokončeno. Po jeho uplynutí je kolo restartováno, avšak hráči je odebrán jeden život. Naopak při rychlém dokončení kola jsou ze zbývajících času vypočteny bonusové body.

Hráči je nabídnuta možnost výběru ze dvou kategorií bonusů.

Do první kategorie patří:

- Život navíc
- Náповěda 50/50 do další otázky
- Zvýšení kapacity zásobníku o jeden

Druhá kategorie je podmíněna velikostí zásobníku a faktem, že hráčův zásobník není zaplněn nevyužitými bonusy. V případě že jsou tyto podmínky splněny, mohou se v zásobníku vyskytnout bonusy v této formě:

- Zrychlení - hráč se může po dobu 5 sekund pohybovat 2x rychleji
- Střela z revolveru - výstřel po zásahu způsobí že strážce zmizí z hracího plánu. Střela letí 2x rychleji než se pohybuje hráč.
- Střela z opakováčky - střela se nezastaví po zásahu strážce, ale až po zásahu stěny. Za každého dalšího zasaženého strážce je dvojnásobný počet bodu.
- Střela z kanónu - výstřel letí jeden a půl krát rychleji než se pohybuje hráč. Střela se za letu odpálí mezerníkem a odstraní všechny strážce v okruhu 3x3.

- Dynamit - po umístění má dynamit časovou prodlevu 3 sekundy než vybuchne a odstraní vše včetně stěn v okruhu 3x3.

Hráč má na počátku hry k dispozici 3 životy a zásobník na zbraně o velikosti 3. Tyto hodnoty se pak můžou v průběhu hry zvyšovat získanými bonusy. Zatímco velikost zásobníku se v průběhu hry může pouze zvětšovat, životy může hráč ztratit výše uvedenými okolnostmi.

Při průchodu úrovní se hráč může setkat s následujícími typy polí:

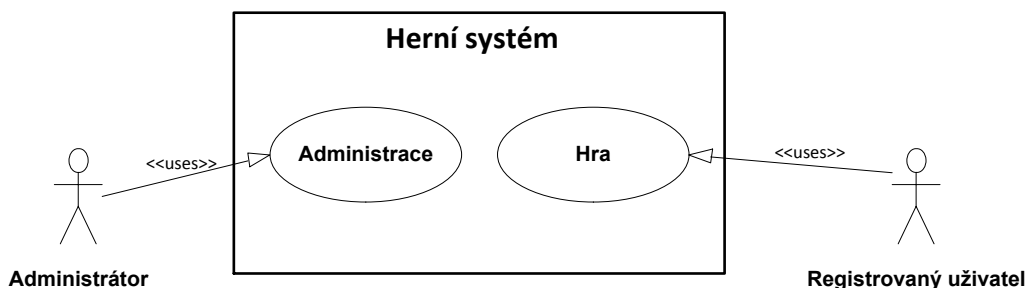
- země - základní pole, na které je možné vstoupit
- počáteční pozice hráče - pole na kterém hráč začíná kolo a na které se vrací při úmrtí. Toto pole se v kole vyskytuje právě jednou a jeho výskyt je podmíněn v definici kola v DB.
- východ z bludiště - vstup na toto pole umožní postup do dalšího kola nebo ukončení hry v případě kola posledního. Může se vyskytovat vícekrát, avšak minimálně jednou. Jeho výskyt je ověřován před uložením kola do DB.
- zeď - je pole, na které není možno vstoupit. Zdi jsou automaticky vytvořeny kolem celé mapy, tvůrce pomoci zdi vytváří bludiště.
- otázka - při vstupu na pole s otázkou je hráči položena otázka ze sady obtížnosti dané tvůrcem kola. Jak již bylo uvedeno, správným zodpovězením otázky hráč obdrží body, bonus a pole otázky se změní v zem. Jelikož jsou otázky vybírány náhodně, je ve hře paměť posledních položených otázek, aby se zabránilo jejich opakování.

Hra je ukončena při ztrátě posledního života, dokončení posledního kola nebo při ukončení hry hráčem. Po tomto ukončení hry je hráčovo skóre zapsáno do výsledkové listiny. Výše skóre je tedy hlavní motivací hráče. Body skóre hráč získává za správně zodpovězené otázky, kde výše bodu je vypočtena podle vzorečku $points = QUESTION_POINTS * QUESTION_LEVEL$, kde $points$ je výsledný počet bodů, $QUESTION_LEVEL$ je obtížnost otázky v rozmezí 1-4 a $QUESTION_POINTS$ je konstanta o hodnotě 10. Další možností jak získat body je zabít strážce, kdy je výše získaných bodů vypočtena podobným vzorcem $point = GUARD_LEVEL * GUARD_POINTS$. $GUARD_LEVEL$ je zde obtížnost strážce, opět v rozmezí 1-4 a $GUARD_POINTS$ je konstanta o velikosti 5. Poslední možností získání bodů jsou bonusové body za rychlost řešení kola. Ty jsou vypočteny pomocí vzorce $bonusPoints = (LEVEL_POINTS * REMAINING_TIME) / LEVEL_TIME$. $LEVEL_POINTS$ je konstanta o velikosti 50. $REMAINING_TIME$ je zbývajíc čas a $LEVEL_TIME$ je celkový čas určený k řešení kola.

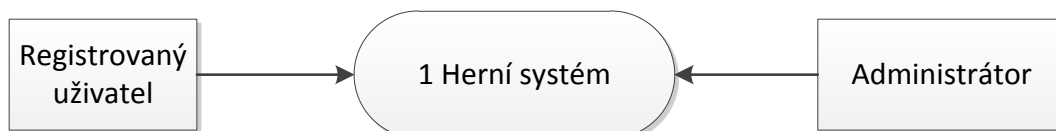
4 Analýza

V této části práce se budu zabývat analýzou požadavků na aplikaci.

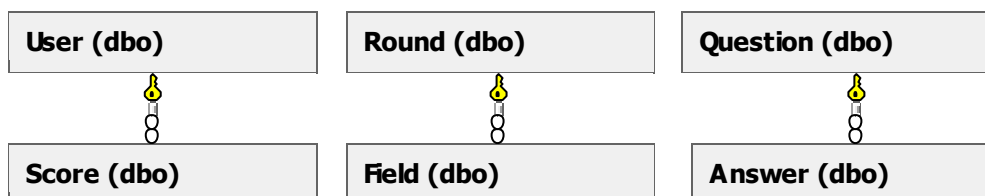
Aplikaci budou moci užívat dva typy uživatelů. Prvním typem bude hráč, který bude moci aplikaci spustit, hrát hru, zobrazit tabulku výsledků za danou dobu a případně změnit své registrační údaje. Dalším typem budou uživatelé s oprávněním pro administraci hry. Tito uživatelé, budou mít tedy oproti hráčům možnost editovat herní úrovně, kladené otázky a registrované hráče. Na základě tohoto rozdělení, se nám nabízí možnost veškerou administraci oddělit od hry samotné. V důsledku toho bude mít do administrace přístup pouze uživatel s administračním oprávněním. Výsledná koncepce systému je zobrazena na obrázku č. 1. Na obrázku č. 3 je pak zjednodušený ER diagram pro nastínění rozdělení databáze na kterou se odkazují v následující části.



Obrázek 1: Analýza koncepce herního systému



Obrázek 2: DFD diagram 0. Úrovně



Obrázek 3: Zjednodušený ER diagram

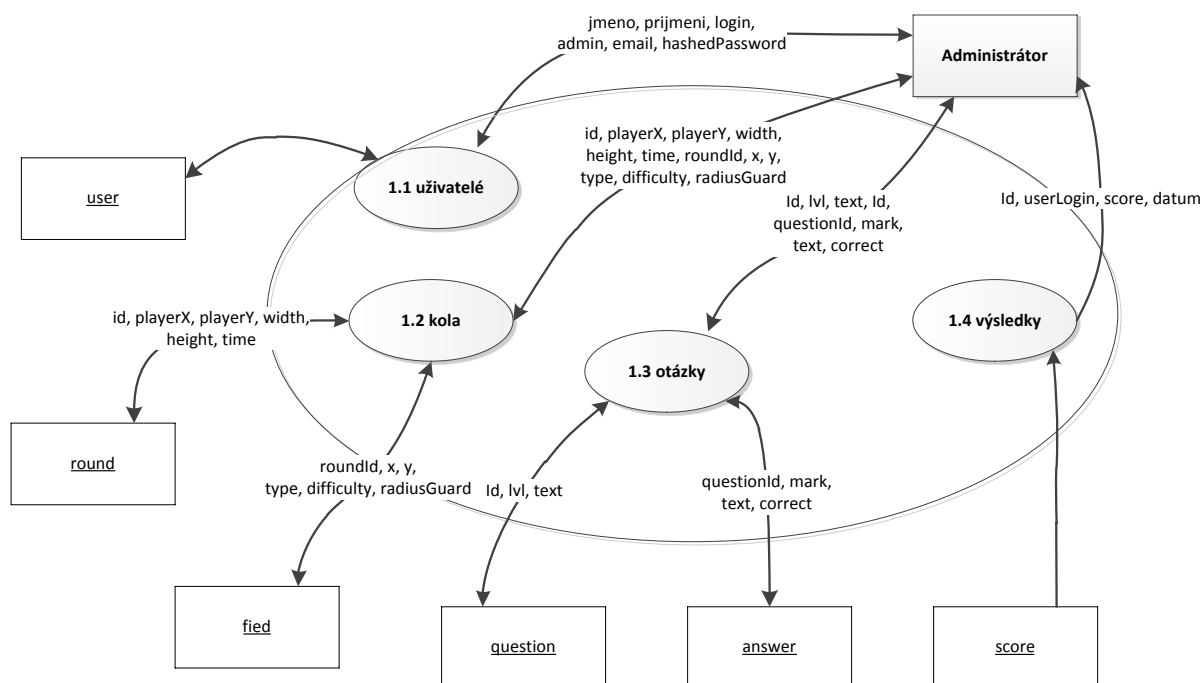
4.1 Administrace

Administrací se rozumí vkládání, editace a případné mazání následujících objektů v databázi:

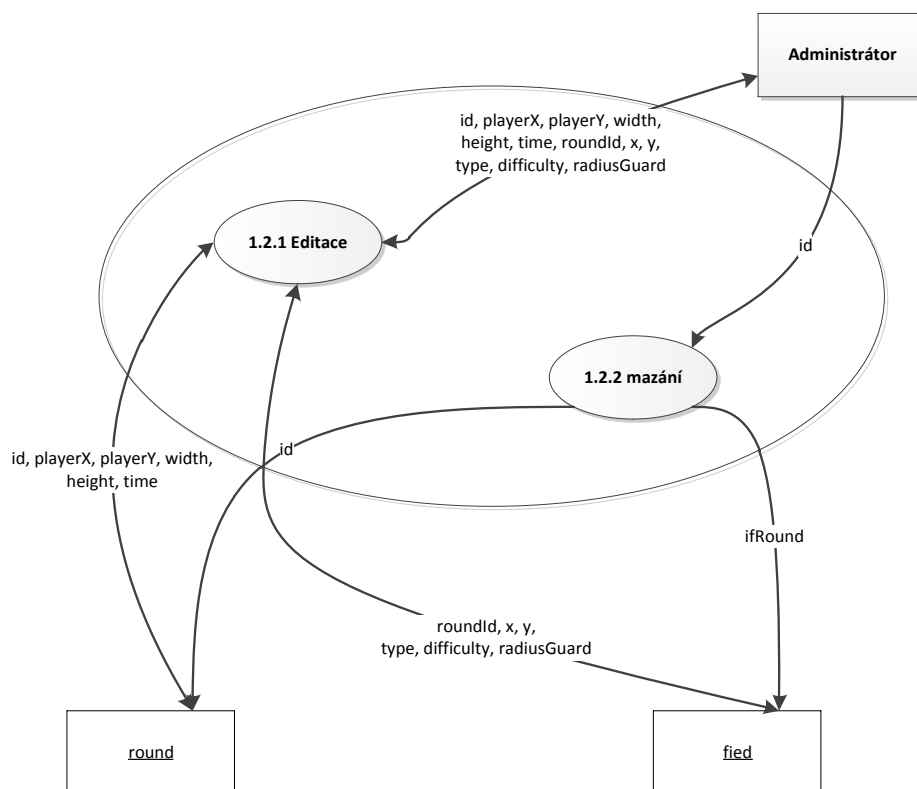
- Prvním typem objektu je uživatel, který je identifikován jednoznačným loginem. U tohoto typu objekty musí být uvedeny také následující údaje: jméno, příjmení, email a heslo. Posledním povinným identifikačním znakem uživatele je jeho role, zda-li je pouze hráč nebo má i administrátorské oprávnění.
- Dalším typem objektu jsou otázky a odpovědi. Každá z otázek je zařazena do jedné ze čtyř úrovní obtížnosti a má definovány čtyři možné odpovědi. Z nichž právě jedna je správná.
- Posledním z objektů je kolo. Definice kola se skládá z šířky a výšky herního plánu, výchozích souřadnic hráče, času poskytnutého ke zdolání kola a definici jednotlivých objektů na herním plánu.

V administraci je také přístupný objekt výsledků. Zde má administrátor pouze dvě možnosti, a to zobrazení nebo smazání výsledků. Vkládání záznamů je možné pouze při ukončení hry samotnou aplikací. Editace záznamů není přístupná vůbec.

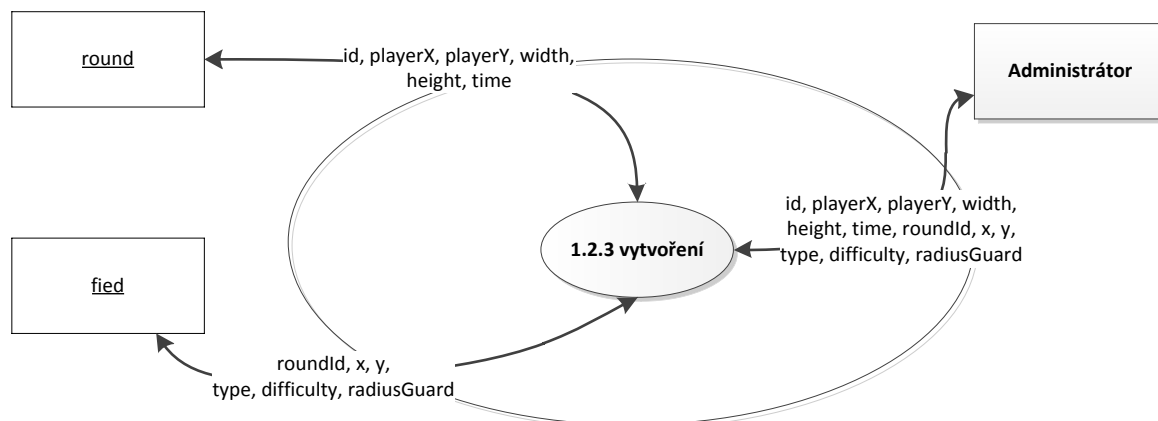
Jednotlivé procesy administrace, které jsou spojeny s prací s daty v DB, jsou zobrazeny v use-case diagramu na obrázku 8. Akce, které je možné vykonávat v administraci jsou zachyceny v DFD diagramu na obr. 4. DFD diagramy na obrázcích č. 5 a 6 zobrazují detailnější náhled na administraci herního kola. Akce administrace otázek a odpovědí je analogická, pouze za použití jiných tabulek databáze a proto DFD diagramy této akce neuvádím.



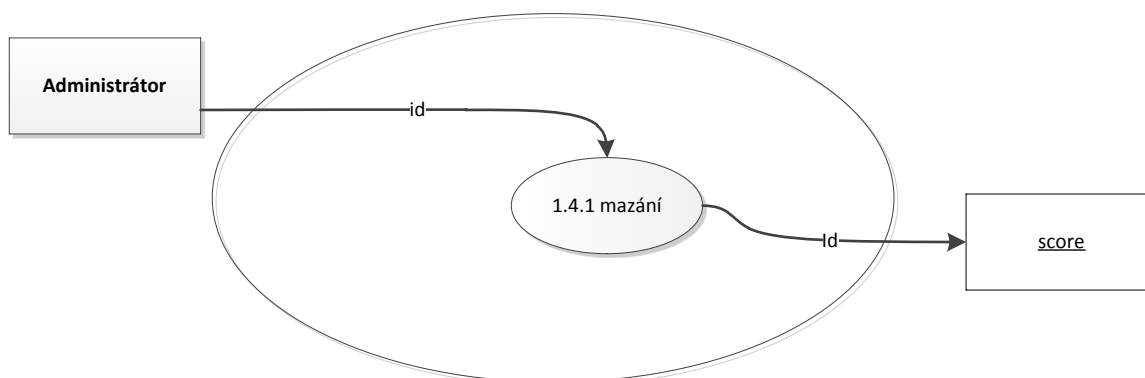
Obrázek 4: DFD diagram 1. Úrovně zobrazující akce přístupné uživateli s administrátorskými právy



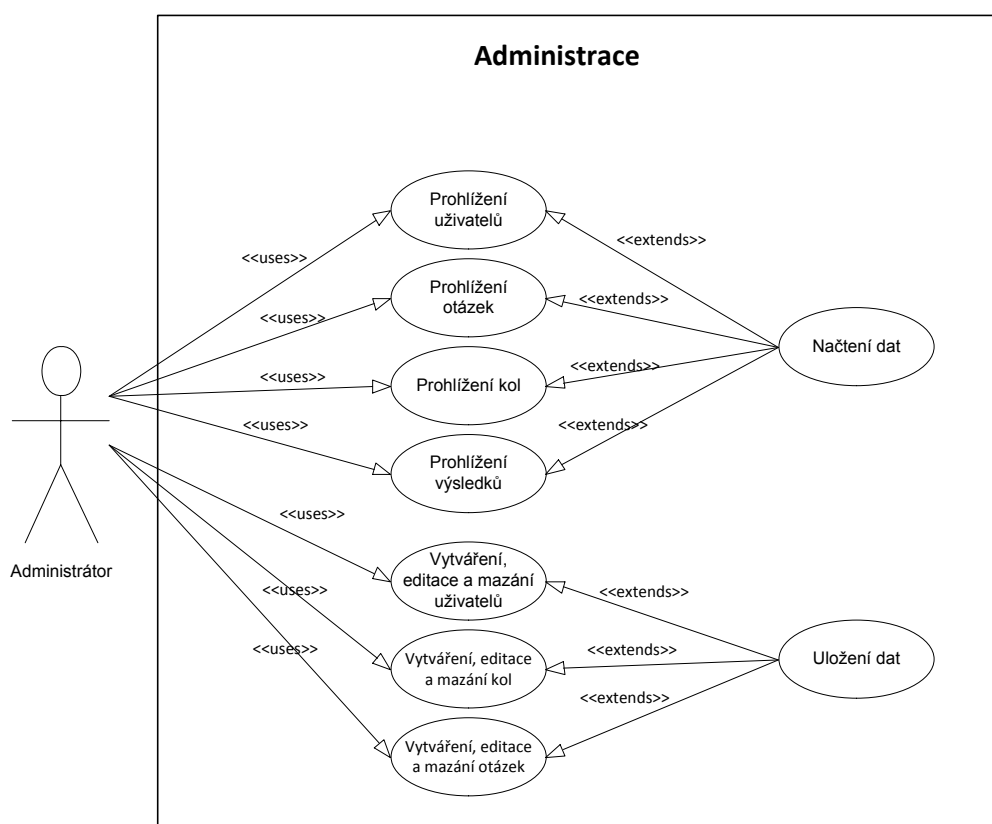
Obrázek 5: DFD diagram 2. Úrovně pro administraci kol (editace a mazání)



Obrázek 6: DFD diagram 2. Úrovně pro administraci kol (vlození)



Obrázek 7: DFD diagram 2. Úrovně pro administraci výsledků

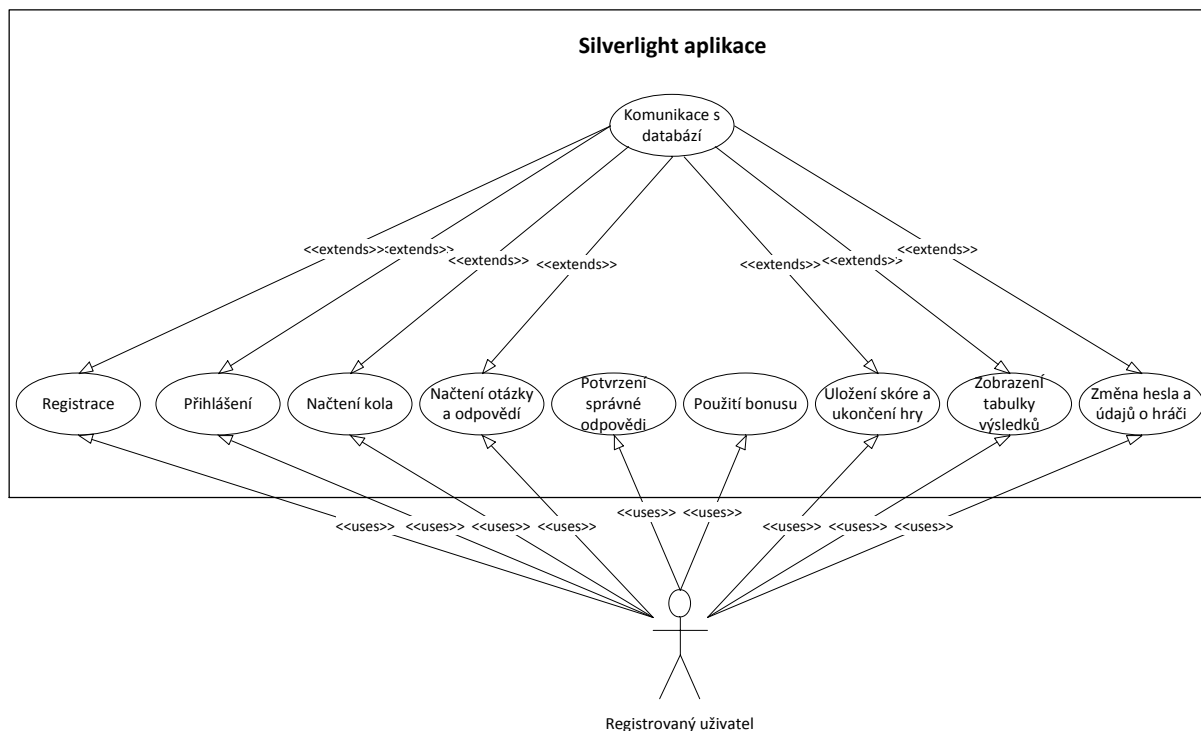


Obrázek 8: Use-case diagram administrace

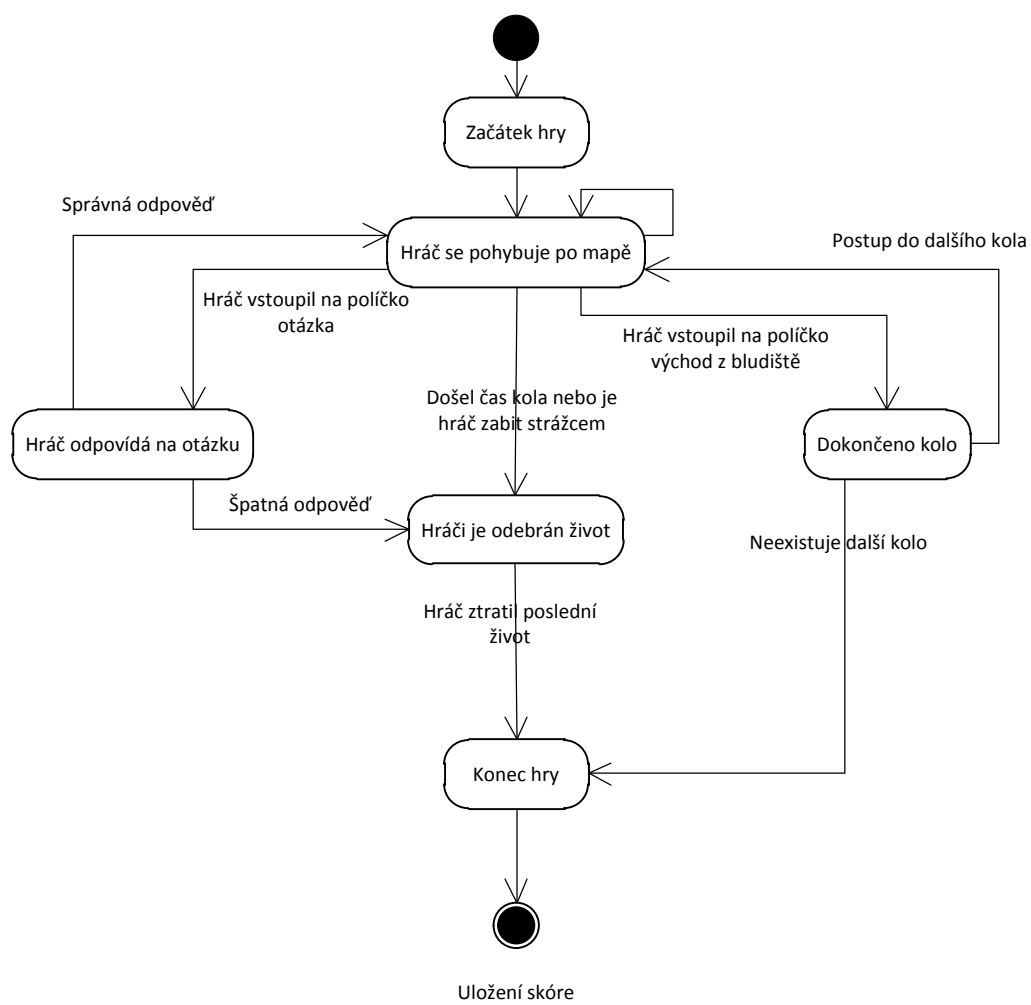
4.2 Hra

Procesy hry jsou zobrazeny na use-case diagramu na obrázku č. 9. Většina z těchto procesů využívá služby WCF pro komunikaci s databází. Pouze procesy ověření správné odpovědi a použití bonusu nevyžadují komunikaci s DB a jsou tedy řešeny přímo herní aplikací.

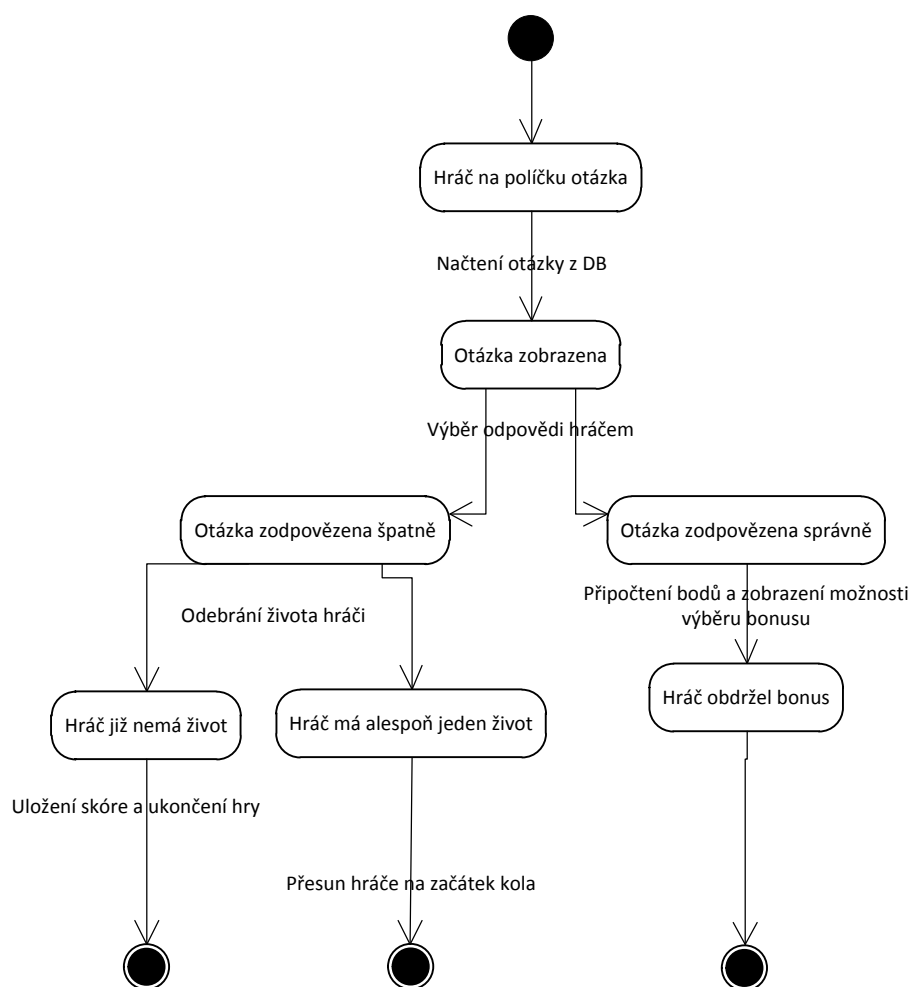
Proces registrace uživatele umožní vytvoření nového uživatele. Po odeslání požadavku je ověřeno, zda-li bylo vložení uživatele do databáze úspěšné. Proces přihlášení vysílá požadavek na ověření uživatelského jména a hesla a očekává odpověď WCF služby, zda-li bylo přihlášení úspěšné. Proces změny hesla a údajů o hráči je obdobný jako proces registrace, pouze s rozdílem, že nekládá, ale jen edituje existující záznam. Procesy načtení kola, načtení otázky a odpovědi a proces zobrazení tabulky výsledů probíhají tak, že hra vyšle požadavek na data, které po jejich obdržení zpracuje. Poslední proces je uložení výsledků. Tento proces při ukončení hry vyšle požadavek na uložení výsledku do databáze.



Obrázek 9: Use-case diagram hry



Obrázek 10: Stavový diagram hry



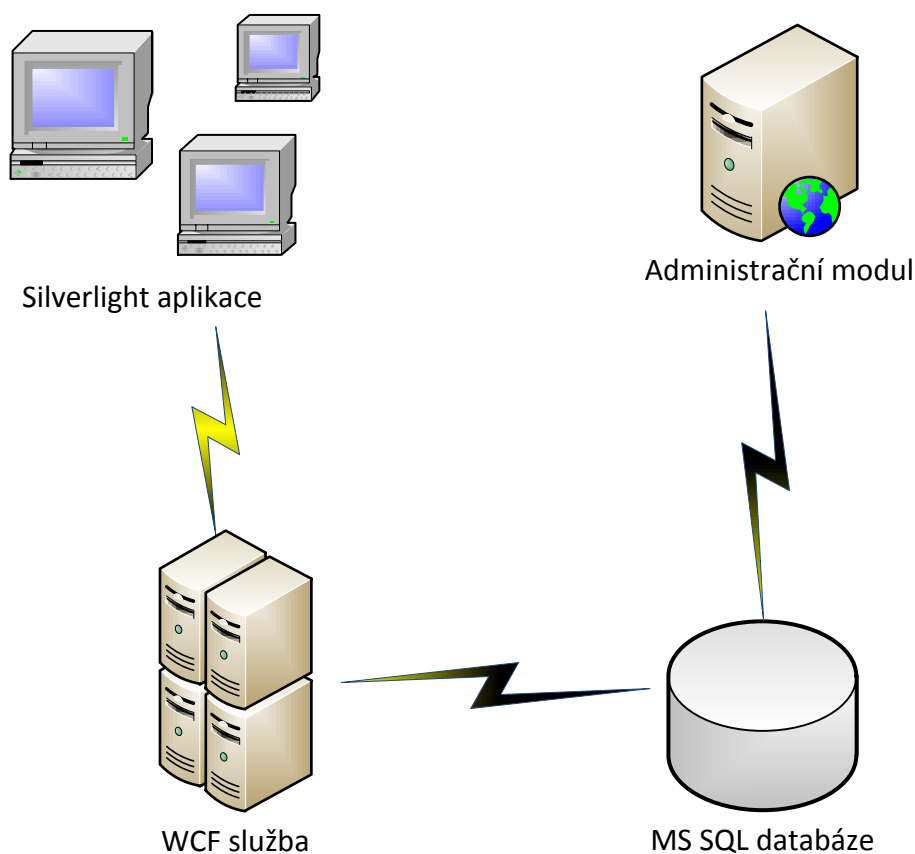
Obrázek 11: Stavový diagram otázky

5 Návrh

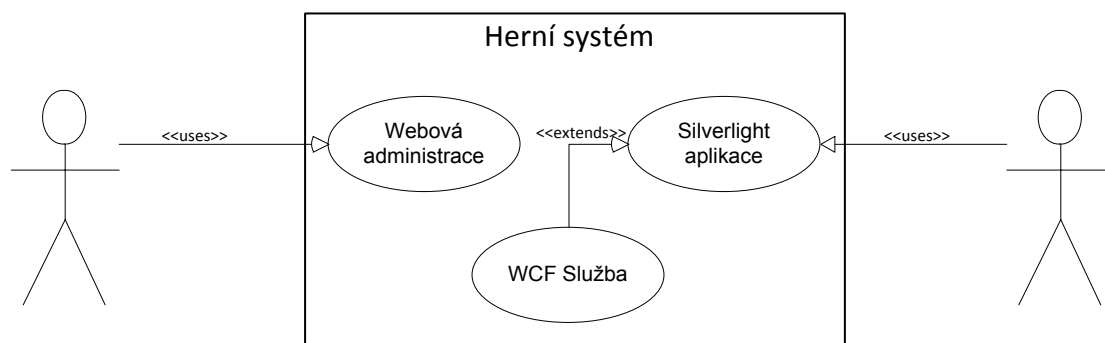
Nyní se budu zabývat návrhem dílčích herních modulů a rád bych také zmínil některé netriviální algoritmy použité v této aplikaci.

5.1 Náhled na celé řešení

Herní aplikace je napsána v Silverlight 4. Tato technologie z bezpečnostních důvodů neumožňuje aplikacím komunikovat přímo s databází. Je tedy zapotřebí použít ještě jednu vrstvu a tou je webová služba. Komunikace s databází tedy probíhá za použití služby WCF. Náhled na řešení aplikace je na obrázku č. 12. Z důvodu použití služby WCF byl poupraven use-case diagram aplikace na obrázku č. 13. Administrační modul je vytvořen za použití ASP.NET, který je často využíván k tvorbě moderních webových prezentací a jako webové rozhraní informačních systémů.



Obrázek 12: Náhled na celé řešení aplikace



Obrázek 13: Use-case diagram aplikace

5.2 Uložení dat

Při tvorbě této aplikace, bylo nutné řešit otázku způsobu uložení dat. Mezi kritéria pro uložení dat patřila bezpečnost, snadná dostupnost dat z aplikace a jednoduchá editace.

První možností, bylo uložení dat v souboru XML a využití podpory XML přímo v aplikaci Silverlight. Tento způsob je však nevyhovující z hlediska zabezpečení - napsáním adresy XML souboru do prohlížeče by se komukoli zobrazila uložená data a získal by tak například údaje o uživatelích, seznam otázek a odpovědí a možnost tyto data pozměnit. Dalším problémem XML je zápis více uživatelů ve stejnou dobu.

Druhou možností bylo uložení dat do databáze. Tato volba se jeví jako optimální, jak z hlediska zabezpečení, tak z hlediska dostupnosti a jednoduchosti správy dat. Vzhledem k použití technologií z rodiny .NET firmy Microsoft jsem se rozhodl využít databázi MS-SQL, která je v .NET aplikacích přímo podporována.

Spojení s databází z herní aplikace je zajištěno službou WCF. Administrační modul přistupuje k databázi za pomoci micro-ORM Dapper, který pro potřeby této aplikace postačuje a je mnohem rychlejší než klasické ORM jako např. LINQ nebo Entity Framework, viz výkonové testy [4]. Riziko SQL injection je ošetřeno použitím parametrizovaných SQL dotazů.

Pro uložení herních úrovní jsem zvolil metodu Master/Detail, kdy pro jeden záznam v primární tabulce (tabulka `round`) existuje v sekundární tabulce (tabulka `field`) několik (typicky mnoho) souvisejících záznamů. Uložení otázek a odpovědí je obdobné. Pro uložení uživatelů nám vystačí pouze jedna tabulka. K rozlišení role uživatele nám vystačí dvoustavový atribut nabývající hodnoty `true` v případě, že má uživatel roli administrátora. Podrobný popis tabulek v databázovém schématu je v příloze č. 1 Vztahy mezi jednotlivými tabulkami jsou znázorněny v ER diagramu na obrázku č. 14.

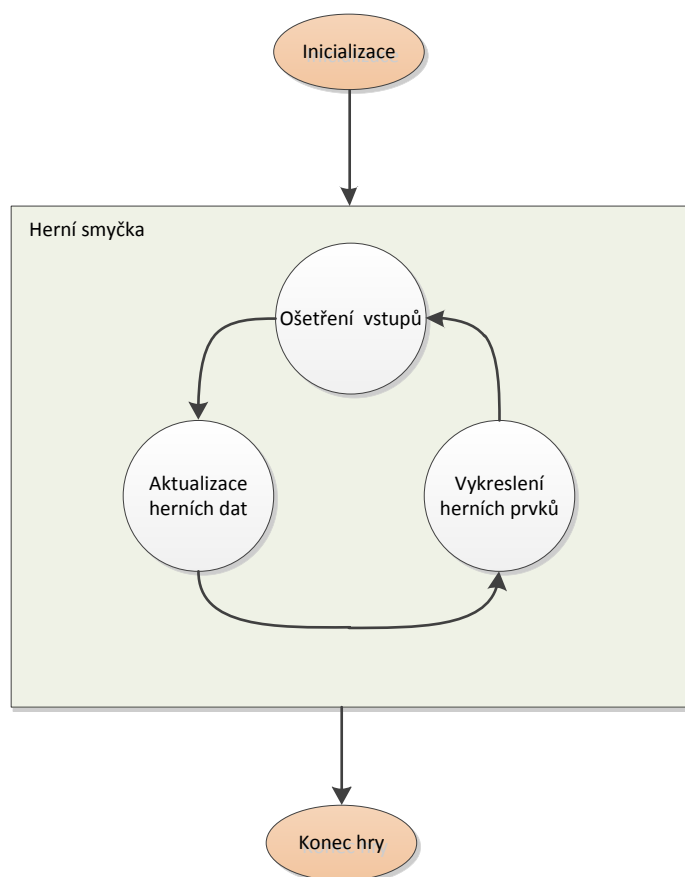


Obrázek 14: ER diagram

5.3 Netriviální algoritmy

5.3.1 Herní smyčka

Funkce GameLoop představuje základní strukturu hry a hlavní bod, ze kterého jsou aktivovány její zbývající části. Po vytvoření okna a inicializaci všech částí hry začne hra vykonávat hlavní smyčku. To je "nekonečný" cyklus, ve kterém se provádí aktualizace herních objektů a vykreslení herního světa. Pro znázornění je na obrázku č. 15. zobrazen průběh herní smyčky.



Obrázek 15: Znázornění herní smyčky

V Silverlightu existuje více možností jak herní smyčku vytvořit. Využil jsem tedy porovnání pěti nejčastějších způsobů tvorby herní smyčky v článku [3]. Pro mé potřeby se ke tvorbě herní smyčky jeví ideální metoda dispatcher timer. Tato metoda dosahuje velmi dobré stability časové délky, což je ideální pro počítání herního času. Dále v rámci této smyčky aktualizuji herní údaje. Uživatelské vstupy jsou ošetřeny použitím událostí, na základě kterých volají smyčky storyboard. Tyto smyčky zajišťují animace herních objektů a ověřování kolizí, viditelnosti a výpočty pro pohyb strážců v bludišti.

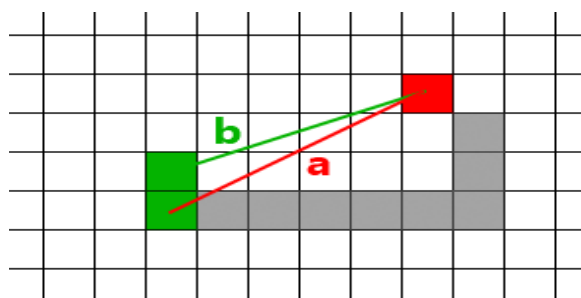
5.3.2 Viditelnost mezi strážcem a hráčem

Strážce typu 1 hlídá prostor určený tvůrcem kola a začíná hráče pronásledovat ve chvíli, kdy je na dohled - mezi hráčem a strážcem není žádná zeď. Při každé změně pozice hráče nebo některého ze strážců typu 1 je tudíž ověřována viditelnost. Bylo tedy nutné najít způsob, jakým toto ověřování bude prováděno. Existuje celá řada algoritmů pro zjištění oblastí viditelnosti ve 2D hrách. Na stránce [1] je porovnávací studie těchto algoritmů. Pro použití v této hře jsou však tyto algoritmy příliš komplexní. V našem případě je dostačující kontrolovat, zda-li se na úsečce mezi hráčem a strážcem nevyskytuje pole zdi.

Prvotní řešení spočívalo ve výpočtu uhlopříček pro každé políčko zdi v oblasti mezi hráčem a strážcem a následná kontrola průsečíku těchto uhlopříček s úsečkou mezi hráčem a strážcem. Počet iterací výpočtů byl tedy přímo úměrný počtu políček zdi v oblasti mezi strážcem a hráčem. V ideálním případě, kdy v této oblasti není žádná zeď, není nutné provést žádný výpočet. Vzhledem k tomu, že předpokládáme herní pole ve formě bludiště, však ve většině případů tento optimální stav nenastane. Ačkoliv byl tento algoritmus rychlostně dostačující, snažil jsem se jej zjednodušit a snížit počet vykonávaných matematických operací.

Nalezl jsem tedy řešení, kdy je nejdříve vypočtena úsečka mezi hráčem a strážcem. Následně zjišťujeme jednotlivá políčka, které tato úsečka protíná ve směru od hráče ke strážci. Je-li některé políčko typu zeď, což je v případě bludiště pravděpodobné, je tento algoritmus ukončen, protože tato zeď viditelnosti mezi hráčem a strážcem zabírá. Počet opakování je přímo úměrný vzdálenosti mezi strážcem a hráčem. Zde je pseudokód kontroly viditelnosti:

```
1. AktualniSouradnice=ObjektA
2. FOR AktualniSouradnice to ObjektB
3.   Zjistí průsečík aktuální souřadnice s úsečkou AB
4.   IF zeď alespoň na jedné straně průsečíku THEN
5.     RETURN viditelnost mezi objektem neexistuje
6.   ENDIF
7.   posun AktualniSouradnice blíž Objektu B
8. ENDFOR
```



Obrázek 16: Příklad kontroly viditelnosti

Na obrázku č. 16 je vyobrazen příklad kontroly viditelnosti - v případě a viditelnost neexistuje jelikož úsečka mezi objekty protíná políčko zdi. V případě b viditelnost mezi objekty existuje.

5.3.3 Hledání nejkratší možné trasy v bludišti.

Zadání hry vyžaduje, aby při vstupu hráče do strážcem střeženého prostoru, tento strážce začal hráče pronásledovat. V okamžiku kdy hráč střežený prostor opustí, se strážce vrací na své původní místo. Vzhledem k tomu, že hrací plán má podobu bludiště, je velká pravděpodobnost, že strážcova trasa nebude přímočará. Je tedy nezbytné najít nejkratší cestu bludištěm. Tímto problémem se zabývá široká škála algoritmů vycházejících z matematického problému teorie grafů. Bludiště si totiž lze představit jako neorientovaný graf, jehož hrany mají stejnou váhu. Hlavním zdrojem informací o algoritmech pro řešení bludiště pro mne byla webová stránka [6] od Waltera D. Pullema.

Rozhodl jsem se použít algoritmus s anglickým názvem Shortest Path Finder. Tento algoritmus je rychlý a můžeme jej použít na jakémkoliv bludišti, které splňuje podmínku, že mezi libovolnými dvěma body existuje alespoň jedna cesta. Paměťová náročnost je přímo úměrná velikosti bludiště, což je v našem případě zanedbatelné, jelikož jsou maximální rozměry bludiště 35x35 polí. Princip algoritmu si můžeme představit následovně - bludiště je z výchozího bodu (strážce) zaplavováno kapalinou. Tato kapalina se šíří rovnoměrně všemi směry (v matematice se vlastně jedná o vyhledávání do šířky). Jakmile je políčko zaplaveno, je uložena informace z kterého políčka bylo zaplaveno. Po dosažení hledaného bodu (hráč nebo výchozí pozice strážce) postupujeme podle uložených informací po jednotlivých políčkách zpět ke strážci. Výsledná trasa je nejkratší (případně jedna z možných nejkratších). V podstatě se jedná o algoritmus A* bez použití heuristiky, takže je každému směru pohybu přiřazena stejná váha.

Protože se při každém pohybu hráče trasa strážce mění, je nutné trasu přepočítávat. Nepotřebuji tedy ukládat celou trasu ale jen následující políčko na které má strážce vstoupit. Algoritmus jsem následovně upravil – začínám vyhledávání od koncového bodu. `BodA` v následujícím pseudokódu tedy představuje hráče. Rovnoměrně ve všech směrech kontroluji sousední políčka a ukládám si políčka, z kterých jsem na ně vstoupil (12. řádek pseudokódu). Jakmile kontrola dorazí na políčko se strážcem (`BodB`) je smyčka `WHILE` ukončena (5. řádek pseudokódu). Proměnná `PředchozíPolíčko` tedy obsahuje pozici, na kterou je třeba strážce přemístit. Poté se výpočet opakuje.

```

1.  NajdiDalšíPohyb(BodA, BodB)
2.  Přiřaď všem políčkům Status=Ready
3.  Vlož výchozí pole (A) do fronty a změň jeho stav na
    Status=Waiting
4.  WHILE fronta není prázdná
5.      IF první pole ve frontě je BodB THEN
6          BREAK
7      ELSE
8          zkontroluj sousední políčka prvního políčka ve frontě
9          IF lze vstoupit na sousední políčko THEN
10             IF sousední políčko je ve stavu Status=Ready THEN
11                 přidej do fronty
12                 PředchozíPolíčko = současné první políčko ve frontě
13                 změň stav na Status=Waiting
14             ENDIF
15         ENDIF
16         Změň stav prvního políčka ve frontě na Status=Processed
17         Odeber políčko z fronty
18     ENDIF
19 ENDWHILE
20 Přesuň se na PosledníPolíčko

```

5.4 Silverlight aplikace

Tato aplikace představuje celou hru včetně její herní logiky. Po otevření aplikace, se uživateli zobrazí přihlašovací formulář. Pokud uživatel ještě nemá uživatelské jméno a heslo, je zde k dispozici formulář pro registraci, ve kterém je možnost si jej vytvořit. Po úspěšném přihlášení, se uživatel dostane do herního menu. Zde si může zobrazit tabulku výsledků za určitou dobu a změnit své údaje uvedené při registraci (výjimkou je login, který je neměnný). Nejdůležitější položkou herního menu je možnost spuštění nové hry. Po zvolení této možnosti se uživateli zobrazí herní prostředí a v průběhu toho co je uživateli zobrazována krátká nápověda jak hru hrát, je v pozadí načteno a zobrazeno první kolo. Základní událostí ve hře, je interakce s uživatelem a reakce na stisk směrových šipek pro pohyb strážce, případně tlačítek pro využití bonusu. Při každé změně pozice hráče či strážce, je ověřována viditelnost mezi hráčem a strážci typu 1. Dále jsou ověřovány kolize hráče se strážci, políčky s otázkami nebo východem z bludiště.

5.5 WCF služba

WCF službu pro Silverlight aplikaci představuje třída WCFService.svc.cs ve které je tato služba implementována. Tato třída obsahuje metody pro komunikaci s databází. Každá metoda, kterou chceme na službě volat, musí být označena Atributem [OperationContract]. Každá aplikace, která tuto službu má využívat, musí obsahovat na tuto službu referenci. Komunikace se službou poté probíhá následovně - Nejdříve nastavíme pomocí event handleru metodu, která se má zavolat, jakmile bude

dokončena práce v metodě na WCF službě. Následně metodu WCF služby asynchronně zavolá. Díky asynchronnímu volání nedojde k zablokování zbytku aplikace. Po dokončení volané metody je tedy spuštěna metoda z event handleru a v ní můžeme například zpracovat volaná data.

Požadavky, které jsou zpracovány WCF službou v této aplikaci:

- vložení uživatele při registraci
- přihlášení - ověřuje heslo oproti záznamu v db, využívá metodu hashování saltování
- editace údajů o uživateli
- získání skóre za danou dobu
- získání definice kola
- získání otázky a seznamu odpovědí
- uložení skóre

5.6 Administrace

Tento modul je napsán v jazyce ASP.NET. Přístup do tohoto modulu mají pouze uživatelé s administrátorským oprávněním (Pokud má atribut `admin` v tabulce `user` hodnotu `true`).

Po úspěšném přihlášení, se uživateli zobrazí v levé části obrazovky menu, ve kterém uživatel může zvolit, se kterými objekty chce pracovat. Po zvolení každé sekce, je uživateli zobrazen gridview s jednotlivými záznamy. Tam kde je to potřebné, má uživatel možnost vložit novou položku nebo editovat či smazat stávající.

Pro přístup k databázi je použita knihovna Dapper, vydaná pod licencí Apache License 2.0. Samotný přístup k datům s použitím Dapperu vypadá následovně:

```
public static IEnumerable<Answer> GetAnswers(int questionId)
{
    string sql = "SELECT id, questionId, mark, text, correct
FROM [Answer] WHERE questionId = @id";

    var answers = MyConnection.Current.Query<Answer>(sql,
new { id = questionId });
    return answers;
}
```

5.6.1 Administrace uživatelů

Tato část administrace, zobrazuje gridview se seznamem uživatelů uložených v tabulce `user`. U každého záznamu, je možnost smazání nebo editace, která zobrazí formulář s konkrétním záznamem. Nad seznamem s uživateli je tlačítko umožňující vložení nového uživatele.

5.6.2 Administrace výsledků

V této sekci se zobrazí gridview s výsledky hry uloženými v tabulce `score`. Záznamy jsou řazeny podle skóre sestupně. Administrátor může tyto záznamy filtrovat podle data vložení a u každého záznamu je k dispozici tlačítko smazat. V sekci skóre není možno údaje vkládat ani editovat.

5.6.3 Administrace otázek

V administraci otázek je podobně jako v administraci uživatelů zobrazeno tlačítko pro vložení nové otázky a gridview s otázkami uloženými v tabulce `questions`. U každého záznamu je opět k dispozici tlačítko editovat a smazat. Oproti administraci uživatelů je zde navíc tlačítko směřující na administraci odpovědí k dané otázce. Tyto odpovědi jsou uloženy v tabulce `answers`.

5.6.4 Administrace kol

Sekce pro administraci kol má na své výchozí stránce opět tlačítko pro přidání úrovně a gridview se seznamem již uložených kol. Volbou přidat novou úroveň se dostaneme na formulář, ve kterém upřesníme základní vlastnosti úrovně - pořadové číslo, výšku a šířku hracího plánu, výchozí souřadnice hráče a čas poskytnutý ke zdolání kola. Po zadání těchto údajů se zobrazí grafické znázornění herního plánu, do kterého administrátor může libovolně umísťovat herní objekty. Před uložením herního plánu je ověřováno, zda-li je na herním plánu alespoň jedno políčko s východem z bludiště.



Obrázek 17: Náhled na rozhraní administračního modulu – návrh kola

6 Řešení

Nyní bych se rád věnoval detailnímu popisu aplikace a také uvedl požadavky pro její úspěšné nasazení.

6.1 Popis aplikace

Jak již bylo zmíněno, aplikace se skládá z více modulů. Běžný uživatel přijde do styku s částí vytvořenou v Silverlightu. Samotná Silverlight aplikace se dále dělí na část definující zobrazení grafického rozhraní a animace psanou pomocí jazyka XAML a kód v pozadí. Tento kód v pozadí je napsán v jazyce C# a zajišťuje veškerou aplikační logiku.

Další služba poskytuje bezpečnou komunikační vrstvu mezi Silverlight aplikací a databází. Obsahuje tedy metody pro databázové dotazy. Výjimkou je pouze třída `PasswordHash`, jež obsahuje metody zajišťující hašování a saltování hesla. Heslo je v databázi uloženo v zahašované a saltované podobě. Touto metodou se snažím minimalizovat možnost získání původní podoby hesla případným útočníkem. Při procesu přihlašování je na uživatelem zadané heslo aplikován stejný algoritmus hashování a saltování. Výsledný řetězec je porovnáván s řetězcem uloženým v DB.

Posledním modulem je administrační rozhraní. Pro vyloučení nutnosti instalace aplikace a pro zvýšení dostupnosti jsem zvolil možnost vytvořit tuto část rovněž jako webovou aplikaci. Byla zvolena technologie ASP.NET, v sekci pro tvorbu kol rozšířená o technologii AJAX zajišťující dynamičnost a uživatelskou atraktivitu.

6.2 Nasazení aplikace

Samotná Silverlight aplikace se spouští na straně klienta, z důvodů použití WCF služby ale hosting aplikace vyžaduje webový server IIS, alespoň ve verzi 7.0. Pro administrační modul je zapotřebí webový server s podporou ASP.NET 4. Pro umístění databáze budeme potřebovat MS SQL Server 2005/2008.

Pro konfiguraci databáze je potřebné v SQL Serveru vytvořit databázové schéma s názvem `gabboDB`. Pro toto schéma by měl být vytvořen uživatel s právy na čtení i zápis do všech tabulek. Tento uživatel bude použit pro přístup WCF služby a administračního modulu do databáze. Následně na tomto schématu spustíme SQL skript `createDB.sql` nacházející se v adresáři `/Aplikace/Install/DB` na přiloženém CD. Tento skript vytvoří všechny tabulky popsané v datovém slovníku v příloze č. 1. Následně tento skript vloží data do těchto tabulek:

- `round` - tři testovací kola hry
- `field` - definice objektů pro tyto kola
- `user` - uživatel Admin s právy administrátora, výchozí heslo pro tohoto uživatele je Administrátor

- question - testovací sada otázek
- answer - odpovědi pro testovací sadu otázek

Nasazení administračního modulu lze provést pouhým nakopírováním dat ze složky /Aplikace/Install/Administrace do složky na webovém serveru. Nesmíme však zapomenout v souboru Web.config nastavit connectionString pro připojení k databázi za použití námi vytvořeného uživatele. Tento connectionString lze vygenerovat například v Microsoft SQL management studiu. Výhodou tohoto konfiguračního souboru je to, že při změně uložště databáze nebo uživatele a jeho hesla není potřebné aplikaci znovu kompilovat. Stačí pouze tyto údaje změnit v tomto souboru za použití jakéhokoliv textového editoru.

Pro zveřejnění samotné hry existují dvě možnosti. První je změnou koncovky souboru BanditGame2.xap v adresáři /Aplikace/Install/Game/ClientBin na koncovku .zip. Následně je potřebné změnit URL adresu WCF služby v souboru ServiceReferences.ClientConfig v tomto zip archivu. Poté je nutné tento archiv přejmenovat zpět na BanditGame2.xap. Další změnou je opět nastavení databázového connectionStringu v souboru /Aplikace/Install/Game/Web.config. Nyní konečně můžeme obsah adresáře /Aplikace/Install/Game nahrát do adresáře hry na webovém serveru.

Druhou, elegantnější možností je použití Visual Studia 2010. Po otevření souboru řešení /Aplikace/Source/Game/BanditGame2/BanditGame2.sln můžeme soubory ServiceReferences.ClientConfig a Web.config editovat přímo ve Visual Studiu 2010. Poté stačí kliknutím pravým tlačítkem myši na projekt BanditGame2.Web zvolit volbu Publish. Po nastavení přihlašovacích údajů k webovému serveru je celá aplikace automaticky publikována.

7 Závěr

Tato bakalářská práce zahrnuje celý proces vývoje online hry. Od specifikace požadavků na aplikaci, analýzy přes návrh až po její řešení a nasazení. Součástí práce je také základní popis použitých technologií, umožňující lepší orientaci v problematice vývoje webových her.

Pro spuštění samotné hry je zapotřebí pouze internetový prohlížeč. Pokud není přítomen interpret Silverlightu, nebo je přítomna jeho starší verze, je automaticky nabídnuto stažení nejnovější verze přímo ze stránek společnosti Microsoft.

Tato online hra byla vytvořena pro podporu výuky databázových předmětů. Je nutno podotknout, že hra je plně funkční a zahrnuje všechny požadované funkce. Je zde však prostor pro zlepšení, jak rozšířením o nové herní prvky a funkce, tak zefektivněním již implementovaných.

Pro možnosti administrace herních kol, otázek a registrovaných uživatelů byl vytvořen samostatný administrační modul. Jelikož je tento modul ve formě webového rozhraní, je pro jeho spuštění nutný pouze internetový prohlížeč.

Při vytváření hry jsem si osvojil použití webových služeb, které rozšiřují funkcionalitu a dynamičnost aplikací. Také jsem se seznámil s některými netriviálními algoritmy, jejichž principy a použití popisují v části zabývající se návrhem.

Poslední část práce obsahuje mimo jiné i instrukce pro úspěšné nasazení a zprovoznění aplikace.

Reference

- [1] Comparative study of field of view algorithms for 2D grid based worlds. *RogueBasin* [online]. 2009 [cit. 2012-08-14]. Dostupné z: http://roguebasin.roguelikedev.com/index.php/Comparative_study_of_field_of_view_algorithms_for_2D_grid_based_worlds
- [2] MIHAYLOV, Nikola. *Anatomy of a Silverlight Game: Easily Create Engaging Online Games*. 2009. Dostupné z: <http://blogs.msdn.com/b/nikola/archive/2009/02/05/anatomy-of-a-silverlight-game-avoid-common-mistakes-when-building-silverlight-online-games.aspx>
- [3] MIHAYLOV, Nikola. *5 Game Loops in Silverlight: Which is the best?*. 2009. Dostupné z: <http://blogs.msdn.com/b/nikola/archive/2009/08/19/exposed-5-methods-to-create-game-loop-which-is-the-best.aspx>
- [4] Dapper - a simple object mapper for .Net. [online]. [cit. 2012-08-12]. Dostupné z: <http://code.google.com/p/dapper-dot-net/>
- [5] HTML5: A vocabulary and associated APIs for HTML and XHTML. [online]. [cit. 2012-08-12]. Dostupné z: <http://www.w3.org/TR/html5/>
- [6] Think Labyrinth!. In: D. PULLEN, Walter. [online]. [cit. 2012-08-12]. Dostupné z: <http://www.astrolog.org/labyrinth/algrithm.htm>
- [7] Základní grafové algoritmy: prohledávání grafů. [online]. [cit. 2012-08-12]. Dostupné z: [http://statnice.obrys.cz/index.php?title=Z%C3%A1kladn%C3%AD_grafov%C3%A9_algoritmy_-_prohled%C3%A1v%C3%A1n%C3%AD_graf%C5%AF_\(do_hloubky,_do_%C5%A1%C3%AD%C5%99ky\),_hled%C3%A1n%C3%AD_nejkrat%C5%A1%C3%AD_cesty,_minim%C3%A1ln%C3%AD_kostra_grafu,_toky_v_s%C3%ADt%C3%ADch](http://statnice.obrys.cz/index.php?title=Z%C3%A1kladn%C3%AD_grafov%C3%A9_algoritmy_-_prohled%C3%A1v%C3%A1n%C3%AD_graf%C5%AF_(do_hloubky,_do_%C5%A1%C3%AD%C5%99ky),_hled%C3%A1n%C3%AD_nejkrat%C5%A1%C3%AD_cesty,_minim%C3%A1ln%C3%AD_kostra_grafu,_toky_v_s%C3%ADt%C3%ADch)

Seznam příloh

Příloha č. 1: Datový slovník

Příloha č. 2: CD se zdrojovými kódy

Adresářová struktura přiloženého CD

/Aplikace/Install	Verze aplikace, která jde spustit (nainstalovat) na webovém serveru
/Aplikace/Source	Zdrojové kódy aplikace
/texty	soubory s textem práce, zadání, klíčová slova a abstrakt

Příloha č. 1: Datový slovník

název	datový typ	délka	PK	FK	povinnost	integritní omezení	popis
jmeno	varchar	20			A		jméno uživatele
prijmeni	varchar	20			A		příjmení uživatele
login	varchar	15	A		A		jednoznačný login uživatele
admin	bit				A		true pokud je administrator
email	varchar	50			A		email
hashedPassword	varchar	200			A		hashované a saltované heslo

Tabulka 1: USER - tabulka s uživateli

název	datový typ	délka	PK	FK	povinnost	integritní omezení	popis
id	int		A		A		
userLogin	varchar	15		A	A		login uživatele
score	int				A		dosažené skóre
datum	smalldatetime				A		datum uložení skóre

Tabulka 2: SCORE - tabulka s výsledky

název	datový typ	délka	PK	FK	povinnost	integritní omezení	popis
id	int		A		A		
playerX	int				A		startovní pozice hráče - souřadnice X
playerY	int				A		startovní pozice hráče - souřadnice Y
width	int				A		šířka herního pole
height	int				A		výška herního pole
time	int				A		čas na kolo v sekundách

Tabulka 3: ROUND - tabulka s atributy kola

název	datový typ	délka	PK	FK	povinnost	integritní omezení	popis
roundId	int		A	A	A		id kola
x	int		A		A		pozice x
y	int		A		A		pozice y
type	int				A		typ pole
difficulty	int						obtížnost - použito u strážce a otázky
radiusGuard	int						poloměr hlídané oblasti = použito u strážce

Tabulka 4: FIELD - tabulka s jednotlivými poličky kola

název	datový typ	délka	PK	FK	povinnost	integritní omezení	popis
id	int		A		A		číslo otázky
lvl	int				A		obtížnost otázky
text	varchar	1024			A		text otázky

Tabulka 5: QUESTION - tabulka znění otázek

název	datový typ	délka	PK	FK	povinnost	integritní omezení	popis
id	int		A		A		
questionId	int			A	A		id otázky
mark	varchar	1			A	a, b, c nebo d	písmeno odpovědi
text	varchar	1024			A		text odpovědi
correct	bit				A		příznak správné odpovědi

Tabulka 6: ANSWER - tabulka možných odpovědí na otázky